

# Physics-Informed Neural Network and its applications


Hung Tran-Nam<sup>1,2</sup>

<sup>1</sup>Institute for Computational Science and Artificial Intelligence,  
Van Lang University, Ho Chi Minh City, Vietnam

<sup>2</sup>Faculty of Fundamental Sciences, Van Lang University, Ho Chi Minh City, Vietnam

Ngày 6 tháng 4 năm 2025

# Contents

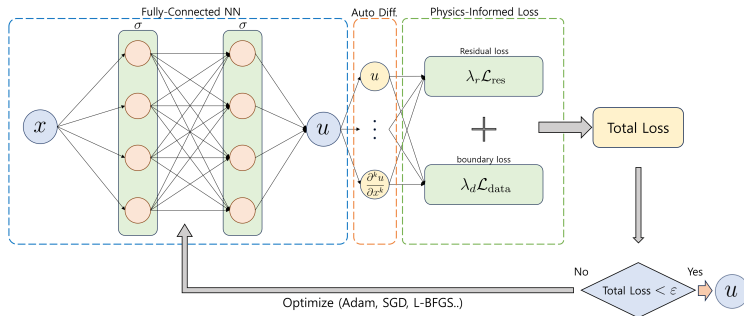
- 
- ① Introduction to PINNs
  - ② Mathematical Foundations
  - ③ Numerical examples
  - ④ Pros and Cons



# Khái niệm

Physics-Informed Neural Network (PINN) là một phương pháp học sâu có giám sát, kết hợp **dữ liệu** và các **định luật** vật lý trong quá trình huấn luyện để xấp xỉ nghiệm của các phương trình vi phân.

- Sử dụng mạng nơ-ron lan truyền thẳng (feed-forward neural network)
- Tích hợp trực tiếp các toán tử vi phân vào hàm mất mát.



## Các ký hiệu

Giả sử miền không gian  $\Omega \subset \mathbb{R}^d$  với biên  $\partial\Omega$ ,

Phương trình vi phân:  $\mathcal{F}(u^*(t, \mathbf{x}); \gamma) = f(t, \mathbf{x}), \quad (t, \mathbf{x}) \in [0, T] \times \Omega,$

Điều kiện biên/ban đầu:  $\mathcal{B}(u^*(t, \mathbf{x})) = g(t, \mathbf{x}), \quad (t, \mathbf{x}) \in [0, T] \times \partial\Omega,$

trong đó:

- $u^*(t, \mathbf{x})$  là nghiệm chính xác, phụ thuộc vào thời gian  $t$  và không gian  $\mathbf{x}$ .
- $\gamma$  là tập các tham số vật lý của hệ.
- $\mathcal{F}$  là toán tử vi phân xác định động lực học của hệ và  $f(t, \mathbf{x})$  là hàm nguồn.
- $\mathcal{B}$  là toán tử điều kiện biên/ban đầu, với  $g(t, \mathbf{x})$  là hàm biên/ban đầu tương ứng.

## 1.1. Mạng nơ-ron truyền thẳng (Feed-forward Neural Network)

Giả sử mạng có  $H$  lớp ẩn, mỗi lớp có  $n_h$  nơ-ron,

$$\hat{u}_\theta(t, \mathbf{x}) = \ell^{(H)} \circ \ell^{(H-1)} \circ \dots \circ \ell^{(1)}(t, \mathbf{x}), \quad (1)$$

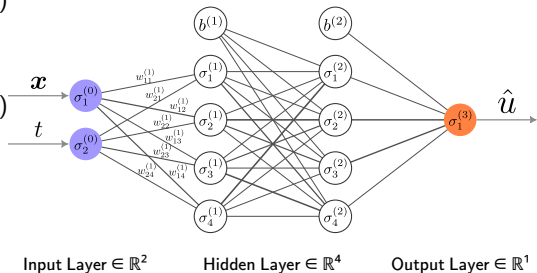
với mỗi lớp  $\ell^{(h)}$  được định nghĩa bởi:

$$\ell^{(h)}(\mathbf{x}) = \sigma^{(h)} \left( \mathbf{W}^{(h)} \mathbf{x} + \mathbf{b}^{(h)} \right), \quad h = 1, \dots, H, \quad (2)$$

trong đó:

- $\mathbf{W}^{(h)} \in \mathbb{R}^{n_h \times n_{h-1}}$  là ma trận trọng số.
- $\mathbf{b}^{(h)} \in \mathbb{R}^{n_h}$  là vector bias.
- $\sigma^{(h)}$  là hàm kích hoạt của lớp  $h$ .

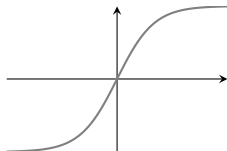
**A fully connected feed-forward neural network**



## 1.2. Một số hàm kích hoạt phổ biến

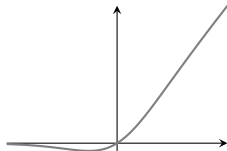
### Tanh

$$\sigma(x) = \tanh(x)$$



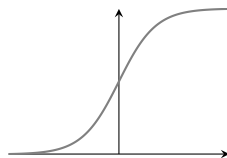
### Swish

$$\sigma(x) = x \cdot \text{sigmoid}(x)$$



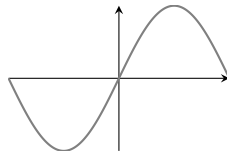
### Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



### Sin

$$\sigma(x) = \sin(x)$$



## 2. Hàm mất mát

Cho  $\hat{u}_\theta$  là nghiệm xấp xỉ của  $u^*$  bằng mạng nơ-ron.

$$\mathcal{L}(\theta) = \omega_{\text{PDE}} \mathcal{L}_{\text{PDE}}(\theta) + \omega_{\text{BC}} \mathcal{L}_{\text{BC}}(\theta) + \omega_{\text{data}} \mathcal{L}_{\text{data}}(\theta) \quad (3)$$

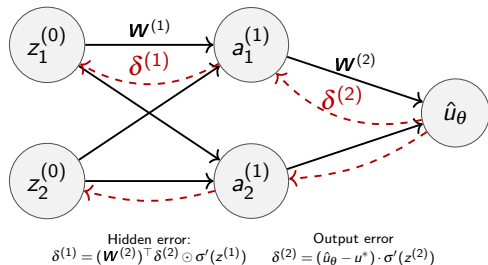
trong đó:

- **Loss phương trình vi phân:**  $\mathcal{L}_{\text{PDE}} = \frac{1}{N_{\text{PDE}}} \sum_{i=1}^{N_{\text{PDE}}} \|\mathcal{F}(\hat{u}_\theta(t_i, \mathbf{x}_i)) - f(t_i, \mathbf{x}_i)\|^2$
- **Loss điều kiện biên/ban đầu:**  $\mathcal{L}_{\text{BC}} = \frac{1}{N_{\text{BC}}} \sum_{i=1}^{N_{\text{BC}}} \|\mathcal{B}(\hat{u}_\theta(t_i, \mathbf{x}_i)) - g(t_i, \mathbf{x}_i)\|^2$
- **Loss dữ liệu:**  $\mathcal{L}_{\text{data}} = \frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} \|\hat{u}_\theta(t_i, \mathbf{x}_i) - u_i^*\|^2$

Mục tiêu tìm  $\theta^* = \{\mathbf{W}^{(h)}, b^{(h)}\}$  tối ưu bằng cách giảm hàm mất mát tổng:

$$\theta^* = \arg \min_{\theta} \mathcal{L}(\theta)$$

### 3. Tối ưu tham số $\theta$ trong PINN



#### 1. Lan truyền sai số:

- Tính sai số lớp đầu ra:

$$\delta^{(2)} = (\hat{u}_\theta(t, x) - u^*(t, x)) \cdot \sigma'(z^{(2)})$$

- Tính sai số lớp ẩn:

$$\delta^{(1)} = (\mathbf{W}^{(2)})^T \delta^{(2)} \odot \sigma'(z^{(1)})$$

#### 2. Tính gradient và cập nhật:

- Gradient lớp đầu ra:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(2)}} = \delta^{(2)} (\mathbf{a}^{(1)})^T, \quad \frac{\partial \mathcal{L}}{\partial \mathbf{b}^{(2)}} = \delta^{(2)}$$

- Gradient lớp ẩn:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(1)}} = \delta^{(1)} (\mathbf{z}^{(0)})^T, \quad \frac{\partial \mathcal{L}}{\partial \mathbf{b}^{(1)}} = \delta^{(1)}$$

- Cập nhật tham số:

$$\mathbf{W}^{(h)} \leftarrow \mathbf{W}^{(h)} - \alpha \frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(h)}}, \quad \mathbf{b}^{(h)} \leftarrow \mathbf{b}^{(h)} - \alpha \frac{\partial \mathcal{L}}{\partial \mathbf{b}^{(h)}},$$

$h = 1, 2$



# Thuật toán huấn luyện PINN

---

**Algorithm 1** Huấn luyện PINN bằng Mini-batch Gradient Descent

---

**Require:** Tập điểm biên  $\{(z_b^i, u_b^{*i})\}_{i=1}^{N_{BC}}$ , tập điểm PDE  $\{z_f^i\}_{i=1}^{N_{PDE}}$ , (tùy chọn) dữ liệu thực  $\{(z_d^i, u_d^{*i})\}_{i=1}^{N_{data}}$

- 1: Khởi tạo mạng nơ-ron với tham số  $\theta$  ngẫu nhiên
  - 2: Thiết lập siêu tham số huấn luyện (số epoch, learning rate  $\alpha$ , dropout)
  - 3: **for** mỗi epoch **do**
  - 4:   Chia dữ liệu thành các mini-batch
  - 5:   **for** mỗi mini-batch **do**
  - 6:     Dự đoán tại điểm biên:  $\hat{u}_\theta(t_b, \mathbf{x}_b)$
  - 7:     Tính phần dư PDE:  $\mathcal{F}(\hat{u}_\theta(t_f, \mathbf{x}_f)) - f(t_f, \mathbf{x}_f)$
  - 8:     Tính hàm mất mát:  $\mathcal{L} = \omega_{PDE} \mathcal{L}_{PDE} + \omega_{BC} \mathcal{L}_{BC} + \omega_{data} \mathcal{L}_{data}$
  - 9:     Tính gradient:  $\nabla_\theta \mathcal{L}$  bằng backpropagation
  - 10:    Cập nhật tham số:  $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}$
  - 11:   **end for**
  - 12: **end for**
-

## Đạo hàm tự động trong PINN

- PINN yêu cầu đạo hàm của  $\hat{u}_\theta(t, \mathbf{x})$  theo  $t, \mathbf{x}$
- Dùng **autograd.grad** để tính các đạo hàm bậc cao

$$\frac{\partial \hat{u}_\theta}{\partial t} = \frac{d\hat{u}_\theta}{dz^{(H)}} \cdot \frac{dz^{(H)}}{dz^{(H-1)}} \cdots \frac{dz^{(1)}}{dt}$$

## Ví dụ 1: PINN cho dao động tắt dần

### Bài toán vật lý

- Mô tả hệ dao động tắt dần:

$$u''(t) + 2\gamma u'(t) + \omega^2 u(t) = 0$$

với  $\gamma = 0.3$  và  $\omega = 3.0$

- Điều kiện ban đầu:

$$u^*(0) = 1, \quad u^{*'}(0) = 0$$

- Nghiệm chính xác:

$$u^*(t) = e^{-\gamma t} \cos(\omega t)$$

### Dữ liệu thực nghiệm:

| ID | $t$   | $u_i^*(\pm 0.01)$ | $u_i^*(\pm 0.05)$ | $u_i^*(\pm 0.1)$ |
|----|-------|-------------------|-------------------|------------------|
| 1  | 0.000 | 0.999             | 0.994             | 0.989            |
| 2  | 0.310 | 0.542             | 0.555             | 0.571            |
| 3  | 0.630 | -0.260            | -0.308            | -0.368           |
| 4  | 1.250 | -0.568            | -0.582            | -0.601           |
| 5  | 1.880 | 0.441             | 0.402             | 0.354            |
| 6  | 2.500 | 0.165             | 0.170             | 0.176            |
| 7  | 2.810 | -0.238            | -0.242            | -0.248           |
| 8  | 3.130 | -0.389            | -0.381            | -0.370           |
| 9  | 3.750 | 0.079             | 0.070             | 0.058            |
| 10 | 4.380 | 0.221             | 0.190             | 0.153            |

# Hàm mất mát trong PINN cho dao động tắt dần

- Tổng hàm mất mát:

$$\mathcal{L} = \omega_{\text{PDE}} \mathcal{L}_{\text{PDE}} + \omega_{\text{BC}} \mathcal{L}_{\text{BC}} + \omega_{\text{data}} \mathcal{L}_{\text{data}}$$

- Thành phần phương trình vi phân:

$$\mathcal{L}_{\text{PDE}} = \|\hat{u}_{\theta}''(t) + 2\gamma\hat{u}_{\theta}'(t) + \omega^2\hat{u}_{\theta}(t)\|^2$$

- Điều kiện ban đầu:

$$\mathcal{L}_{\text{BC}} = \|\hat{u}_{\theta}(0) - 1\|^2 + \|\hat{u}_{\theta}'(0) - 0\|^2$$

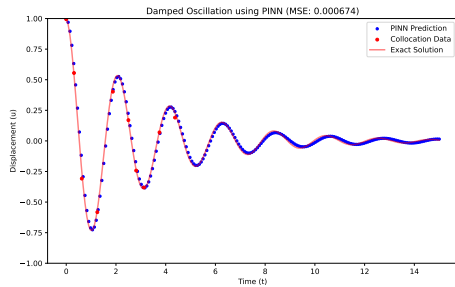
- So sánh với dữ liệu thu thập được:

$$\mathcal{L}_{\text{data}} = \|\hat{u}_{\theta}(t_i) - u_i^*\|^2, i = 1, \dots, 10$$

- So sánh khả năng học của PINN qua các hàm kích hoạt khác nhau;
- So sánh PINN khi data có nhiễu 1%, 5% và 10%;
- So sánh PINN và FCN khi có data có nhiễu 1%, 5% và 10%;
- So sánh PINN và các phương pháp Euler, Improved Euler, RK2 và RK4.

# Kết quả: Các phương pháp Deep Learning

## PINN



```
{ "activation_name": "sine",  
  "num_hidden_layers": 6,  
  "dropout_rate": 0.0,  
  "num_neurons": 64,  
  "num_epochs": 10000,  
  "learning_rate": 0.001,  
  "weight_decay": 0.0001,  
  "omega_eq": 1.0,  
  "omega_bc": 1.0,  
  "omega_dt": 1.0}
```

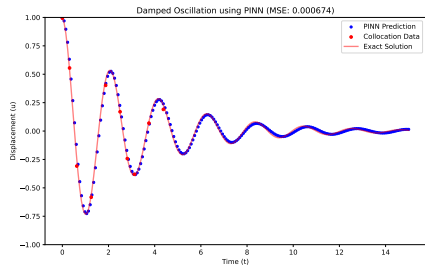
| Kích hoạt        | Noise 1% | Noise 5% | Noise 10% |
|------------------|----------|----------|-----------|
| <b>PINN sine</b> | 0.000680 | 0.000674 | 0.000732  |
| PINN swish       | 0.001093 | 0.001315 | 0.001078  |
| PINN tanh        | 0.017900 | 0.017136 | 0.017242  |
| PINN sigmoid     | 0.058383 | 0.058381 | 0.058378  |

## Tổng hợp kết quả

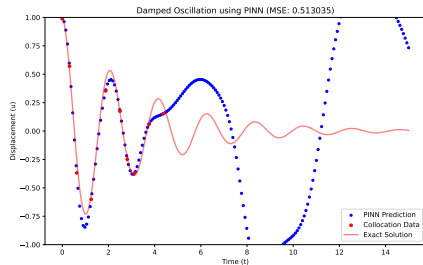
| Phương pháp    | Kích hoạt | Noise 1% | Noise 5% | Noise 10% |
|----------------|-----------|----------|----------|-----------|
| PINN           | sine      | 0.000680 | 0.000674 | 0.000732  |
|                | swish     | 0.001093 | 0.001315 | 0.001078  |
|                | tanh      | 0.017900 | 0.017136 | 0.017242  |
|                | sigmoid   | 0.058383 | 0.058381 | 0.058378  |
| FCN            | sine      | 0.493524 | 0.426112 | 0.513035  |
|                | swish     | 2.754098 | 8.398537 | 9.707447  |
|                | tanh      | 0.115148 | 0.034842 | 0.598950  |
|                | sigmoid   | 0.067203 | 0.065018 | 0.062720  |
| Euler          |           |          | 9.708041 |           |
| Improved Euler |           |          | 0.000266 |           |
| Runge Kutta 2  |           |          | 0.000266 |           |
| Runge Kutta 4  |           |          | 0.000907 |           |

# PINN vs FCN

Noise 5%



(a) PINN



(b) FCN

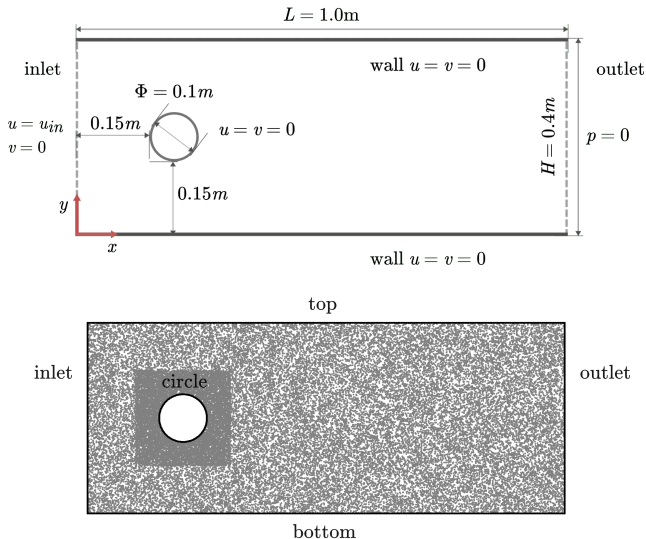


## Ví dụ 2: PINN cho dòng qua vật cản (hình trụ tròn)

- Miền dòng: hình chữ nhật kích thước  $1.0\text{m} \times 0.4\text{m}$ .
- Vật cản là hình trụ tròn, đường kính  $\Phi = 0.1\text{m}$ , tâm tại  $(0.2\text{m}, 0.2\text{m})$ .
- Biên trên và dưới:  $\mathbf{u}^*(z) = 0$  với  $\mathbf{z} = (x, y)$ .
- Lối vào: vận tốc parabol tổng hợp

$$u_x^*(0, y) = \frac{4y(H-y)}{H^2}, \quad u_y^*(0, y) = 0,$$

- Lối ra: áp suất tổng hợp  $p^*(x = 1.0, y) = 0$ .
- **Không sử dụng dữ liệu thực nghiệm** — điều kiện biên đều được tổng hợp từ vật lý.



# Về phương trình Navier–Stokes

- Phương trình Navier–Stokes mô tả chuyển động của dòng chất lỏng (khí hoặc lỏng), kết hợp định luật bảo toàn khối lượng và động lượng:

(1) Bảo toàn khối lượng (dòng không nén):

$$\nabla \cdot \mathbf{u}^* = 0$$

(2) Bảo toàn động lượng:

$$\rho \left( \frac{\partial \mathbf{u}^*}{\partial t} + \mathbf{u}^* \cdot \nabla \mathbf{u}^* \right) = -\nabla p^* + \mu \nabla^2 \mathbf{u}^*$$

- Với trường vận tốc  $\mathbf{u}^* = (u_x^*, u_y^*)$ , áp suất  $p^*$ , khối lượng riêng  $\rho$ , và độ nhớt động lực học  $\mu$ .
- Trường hợp trong bài toán này giả sử dòng không nén và ổn định ( $\partial_t \mathbf{u}^* = 0$ ).

# PINN cho bài toán phương trình Navier–Stokes

- Với miền không gian  $\Omega \subset \mathbb{R}^2$ , mục tiêu là xấp xỉ các trường  $\mathbf{u}^*(\mathbf{z}) = (u_x^*, u_y^*)$ ,  $p^*(\mathbf{z})$ ,  $\mathbf{z} = (x, y)$  thoả mãn hệ phương trình Navier–Stokes trong miền chất lỏng không nén.
- Mạng nơ-ron sinh ra các đầu ra  $\hat{\mathbf{u}}_\theta(\mathbf{z}) = (\hat{u}_x, \hat{u}_y)$ ,  $\hat{p}_\theta(\mathbf{z})$ ,  $\hat{\sigma}_{xx}(\mathbf{z})$ ,  $\hat{\sigma}_{xy}(\mathbf{z})$ ,  $\hat{\sigma}_{yy}(\mathbf{z})$
- PINN sử dụng các phần dư vật lý:

$$\mathcal{R}_1 = \rho(\hat{u}_x \partial_x \hat{u}_x + \hat{u}_y \partial_y \hat{u}_x) - \partial_x \hat{\sigma}_{xx} - \partial_y \hat{\sigma}_{xy}$$

$$\mathcal{R}_2 = \rho(\hat{u}_x \partial_x \hat{u}_y + \hat{u}_y \partial_y \hat{u}_y) - \partial_x \hat{\sigma}_{xy} - \partial_y \hat{\sigma}_{yy}$$

$$\mathcal{R}_3 = -\hat{p}_\theta + 2\mu \partial_x \hat{u}_x - \hat{\sigma}_{xx}$$

$$\mathcal{R}_4 = -\hat{p}_\theta + 2\mu \partial_y \hat{u}_y - \hat{\sigma}_{yy}$$

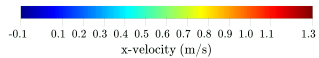
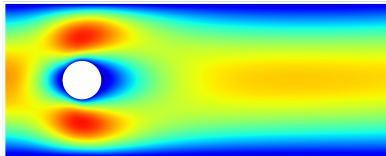
$$\mathcal{R}_5 = \mu(\partial_y \hat{u}_x + \partial_x \hat{u}_y) - \hat{\sigma}_{xy}$$

$$\mathcal{R}_6 = \partial_x \hat{u}_x + \partial_y \hat{u}_y$$

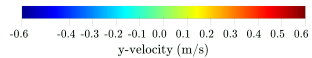
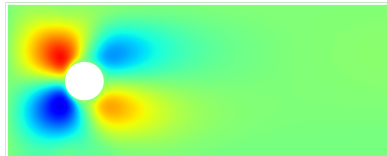
- Hàm mất mát tổng:

$$\mathcal{L}(\theta) = \underbrace{\mathcal{L}_{\text{PDE}}}_{\text{phần dư } \mathcal{R}_1 \rightarrow \mathcal{R}_6} + \underbrace{\mathcal{L}_{\text{BC}}}_{\text{vận tốc tại biên}} + \underbrace{\mathcal{L}_{\text{outlet}}}_{\text{áp suất tại lối ra}}$$

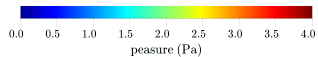
# Kết quả



(c)  $u_x$



(d)  $u_y$



(e)  $p$

## FCN:

- Cần lượng lớn dữ liệu gán nhãn,
- Không đảm bảo tuân thủ các định luật vật lý,
- Dự đoán không ổn định ngoài phân bố training,
- Hiệu suất giảm khi dữ liệu nhiều hoặc thiếu.

## PINN:

- Tận dụng kiến thức vật lý để bù đắp cho thiếu hụt dữ liệu
- Có thể huấn luyện với ít dữ liệu hoặc không có dữ liệu.
- Tích hợp kiến thức PDE chính xác.

# THANK YOU

FOR PARTICIPATING

Hung Tran-Nam